

Absolvování individuální odborné praxe

Individual Professional Practice in a Company

Jakub Riško

Bakalářská práce

Vedoucí práce: Ing. Tomáš Pavelek, Ph.D.

Konzultant práce: Ing. Libor Václavek

Ostrava, 2021

Abstrakt

V textu této bakalářské práce se zabývám popisem mé činnosti v rámci individuální odborné praxe ve firmě Roper Engineering s.r.o. se sídlem v Ostravě. V úvodu vysvětlím své důvody ke zvolení odborné praxe jako hlavní aktivity bakalářské práce. V další kapitolách popíšu odborné zaměření firmy, zadání mých úkolů a jejich řešení. Dále uvedu praktické znalosti a dovednosti, které jsem ve firmě během své padesátidenní praxe získal. Ke konci zhodnotím úspěšnost mé činnosti a uvedu reálnou aplikaci mé tvorby.

Klíčová slova

ATmega328P; Individuální odborná praxe; Měření elektrického proudu; Měření teploty a vlhkosti; Mikrokontrolér; OrCAD; Testovací zařízení

Abstract

In the text of this bachelor thesis, I deal with the description of my activities within the Individual Professional Practice at Roper Engineering s.r.o. located in Ostrava. In the introduction I will explain my reasons for choosing professional practice as the main activity of the bachelor's thesis. In the next chapters I am going to describe the company's specialization, the assignment of my tasks and their solutions. I will also write about the practical knowledge and skills that I have acquired in the company during my fifty-day practice. At the end, I will evaluate the success of my work and present a real application of my work.

Key words

ATmega328P; Individual Professional Practice in a Company; Electric current measurement; Temperature and humidity measurement; Microcontroller; OrCad; Tester

Poděkování

Rád bych poděkoval zástupcům firmy Roper Engineering s.r.o. za příležitost vykonat svou bakalářskou práci formou odborné praxe. Především bych chtěl poděkovat Bc. Jiřímu Kučerovi a Ing. Liborovi Václavkovi za skvělé vedení a ochotu, kterou mi v průběhu mé praxe věnovali.

Dále bych rád poděkoval vedoucímu mé práce Ing. Tomáši Pavelkovi, Ph.D. za konzultace v průběhu vytváření mé bakalářské práce.

Obsah

Seznam použitých zkratek.....	- 6 -
Seznam použitých symbolů.....	- 7 -
Seznam použitých termínů.....	- 8 -
Seznam ilustrací.....	- 9 -
Seznam tabulek	- 10 -
Úvod	- 11 -
1 Odborné zaměření firmy a popis mého pracovního zařazení	- 12 -
1.1 Roper Engineering s.r.o.	- 12 -
1.2 Zařazení a pracovní činnost.....	- 12 -
2 Zadané úkoly, jejich časová náročnost a řešení	- 13 -
2.1 Kreslení schématu v OrCADu dle fyzické předlohy	- 13 -
2.2 Teplotní a vlhkostní senzor.....	- 15 -
2.3 Joystick tester	- 19 -
2.4 Foil tester.....	- 21 -
2.4.1 První verze Foil testeru	- 23 -
2.4.2 Finální verze Foil testeru	- 25 -
2.5 LED indikátor výkonu.....	- 30 -
3 Teoretické a praktické znalosti a dovednosti získané v průběhu studia uplatněné během odborné praxe.....	- 32 -
4 Znalosti či dovednosti scházející studentovi v průběhu odborné praxe	- 33 -
5 Dosažené výsledky v průběhu odborné praxe a její celkové zhodnocení.....	- 34 -
Použitá literatura.....	- 35 -
Seznam příloh.....	- 36 -

Seznam použitých zkratek

Zkratka	Význam
CT	Current transformer
EEPROM	Electrically Erasable Programmable Read-Only Memory
HW	Hardware
LED	Light emitting diode
MCU	Microcontroller unit
PCB	Printed circuit board
RTC	Real time clock
SD	Secure Digital

Seznam použitých symbolů

Symbol	Jednotky	Význam symbolu
I	A	Proud
P	W	Výkon
R	Ω	Odpor
U	V	Napětí
Z	Ω	Impedance

Seznam použitých termínů

Termín	Význam termínu
Arduino IDE	Vývojové prostředí využívající platformy Wiring k programování mikrokontrolérů Arduino
Joystick	Dvouosá či tříosá páka sloužící pro přesné ovládání například vodních trysek
Mikrokontrolér	Integrovaný obvod sdružující do jednoho čipu mikroprocesor a jeho periferie

Seznam ilustrací

<i>Obrázek 1:</i>	<i>Tester univerzálních řídících desek</i>	<i>- 14 -</i>
<i>Obrázek 2:</i>	<i>Kovová podložka řezací desky a kryt krokového motoru.....</i>	<i>- 15 -</i>
<i>Obrázek 3:</i>	<i>Funkční Joystick tester</i>	<i>- 20 -</i>
<i>Obrázek 4:</i>	<i>Fólie používaná pro ovládání řezacího stroje</i>	<i>- 22 -</i>
<i>Obrázek 5:</i>	<i>Schéma zapojení a obrázek fólie Lavamin.....</i>	<i>- 28 -</i>
<i>Obrázek 6:</i>	<i>Funkční Foil tester.....</i>	<i>- 29 -</i>
<i>Obrázek 7:</i>	<i>Foil tester ve vytisknuté plastové krabičce</i>	<i>- 29 -</i>
<i>Obrázek 8:</i>	<i>Schéma zapojení mikrokontroléru pro měření střídavého proudu.....</i>	<i>- 30 -</i>
<i>Obrázek 9:</i>	<i>Nedokončený LED indikátor výkonu</i>	<i>- 31 -</i>

Seznam tabulek

<i>Tabulka 1:</i>	<i>Zapojení pinů Joysticku.....</i>	<i>- 19 -</i>
<i>Tabulka 2:</i>	<i>Příklad uspořádání tlačítek fólie.....</i>	<i>- 21 -</i>

Úvod

Absolvování individuální odborné praxe jsem si zvolil především proto, abych získal praktické znalosti a zkušenosti v oboru, který studuji a ve kterém budu velmi pravděpodobně pracovat celý život. Praktická činnost je totiž, dle mého názoru, nejrychlejší způsob učení se novým dovednostem. Tato metoda má samozřejmě své nevýhody, protože se člověk neučí dle normovaných školních osnov, a proto i postupy, kterým se přiučí nemusí jednoznačně odpovídat těm školním.

Většina obsahu mé bakalářské práce se bude týkat popisu zadání mých úkolů ve firmě, jejich zpracování a zhodnocení užitečnosti. Dále se budu věnovat praktickým znalostem a dovednostem získaným v průběhu mého studia na střední i vysoké škole, které jsem během individuální odborné praxe využil, ale i těm, jež my scházely. Na závěr zhodnotím průběh celé odborné praxe a její přínos.

1 Odborné zaměření firmy a popis mého pracovního zařazení

1.1 Roper Engineering s.r.o.

Firma Roper Engineering s.r.o. sídlí v Ostravě a specializuje se na vývoj a výrobu zařízení pro řezání, čištění, broušení, leštění a leptání metalografických vzorků [\[1\]](#). Jejím hlavním odběratelem je sesterská společnost Struers Inc. se sídlem v Dánsku. Struers je světovou špičkou v oblasti kompletního řešení přípravy a analýzy metalografických vzorků. Firma Roper Engineering je vlastněna společností Roper Technologies Inc. se sídlem na Floridě ve Spojených státech amerických. Ta vlastní další firmy v různých oblastech průmyslu a technologií se zaměřením na specializovaný trh.

Budova Roper Engineering je menší kopií dánského sídla Struers. Budova v Ostravě se dělí na dvě části. První část se skládá především z kancelářského prostoru. Zde se pod značkou Struers vyvíjí nové a spravují již vyráběné zařízení pro zpracování metalografických vzorků. V druhé části budovy je prostorná výrobní hala, kde se pod značkou Struers vyrábí zařízení jako Accutom, Discotom, Secotom nebo například Lavamin [\[2\]](#).

1.2 Zařazení a pracovní činnost

Má pozice ve firmě se jmenuje HW Trainee. Mým hlavním úkolem je pomáhat HW inženýrovi, kterým je nyní Bc. Jiří Kučera, s řešením některých jeho činností a pod jeho dohledem pracovat na samostatných projektech. HW inženýr je odpovědný za optimalizaci a bezpečnost výrobních procesů. Dále se stará se o to, aby se nepoužívaly zastaralé komponenty a součástky a v neposlední řadě analyzuje nahlášené chyby a navrhuje jejich řešení.

Jako můj konzultant je však uveden Ing. Libor Václavek, který ve firmě pracuje jako manažer sekce Engineeringu. Důvodem je to, že pan Bc. Jiří Kučera přestane v průběhu dubna letošního roku z osobních důvodů ve firmě pracovat. Jeho pozici ve firmě nahradí Ing. Denis Hofman, který ve firmě pracuje od začátku roku nový HW inženýr.

Ve firmě jsem začal pracovat od srpna roku 2020. Během výuky v zimním i letním semestru jsem firmu navštěvoval dvakrát týdně, 8 až 9 hodin denně. Formulář s potvrzením o počtu odpracovaných hodin jsem zaslal elektronicky vedoucímu mé bakalářské práce Ing. Tomáši Pavelkovi Ph.D.

2 Zadané úkoly, jejich časová náročnost a řešení

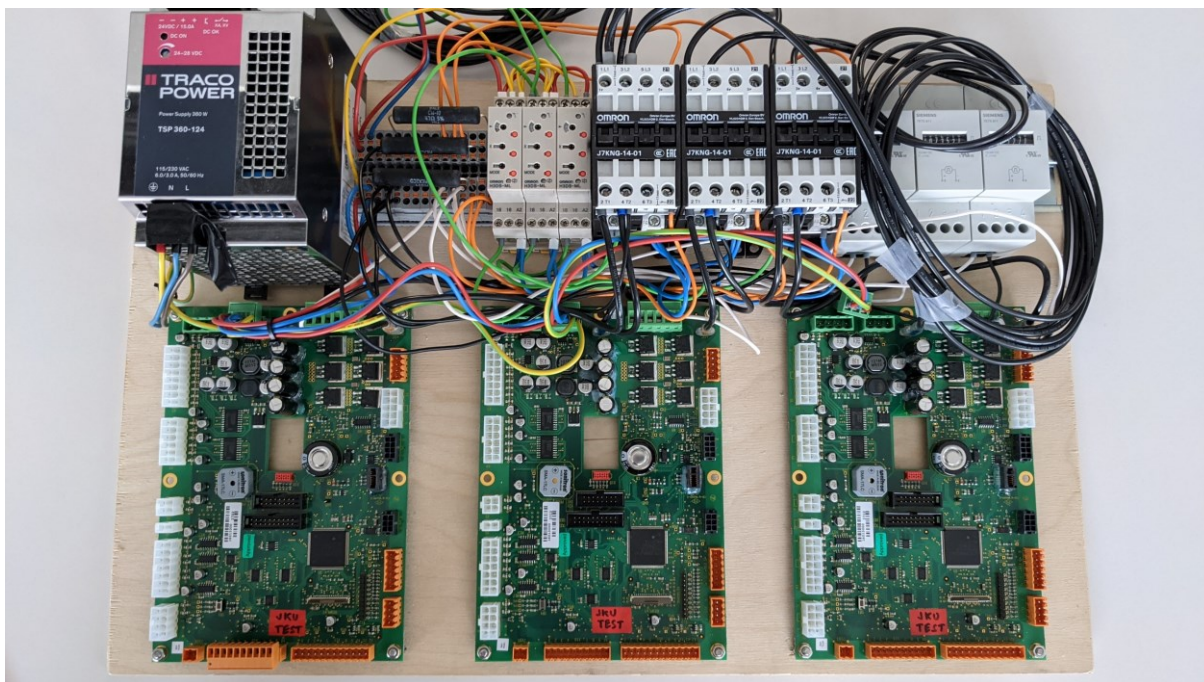
Úkoly, které jsem dostával, souvisely s testováním přístrojů vyráběných ve firmě. Zpočátku většina úkolů souvisela s dokreslením chybějících schémat, nebo přizpůsobením starších schémat aktuálnímu stavu. Další zadané úkoly se týkaly vytvoření testovacích zařízení na bázi mikrokontroléru. Během mé praxe jsem také vyráběl konektory a snímače, které jsem přidával do už vyrobených přístrojů nebo testovacích zařízení v rámci bezpečnostních a jiných modernizací. O tom však psát nebudu, protože se nejednalo o mou vývojovou činnost a zároveň by mohlo jít o nezveřejnitelné firemní informace. Časová náročnost jednotlivých úkolů trvala obvykle dva týdny až několik měsíců.

2.1 Kreslení schématu v OrCADu dle fyzické předlohy

Pro kreslení elektrických schémat se ve firmě využívá program OrCAD Capture. Elektrické schéma musí mít zakreslen každý vyráběný přístroj, samostatný prototyp i testovací zařízení. Prvky ve schématech musí být označeny anglickými názvy. Důvodem je to, že jsou všechna schémata dostupná ve firemní online databázi, kde k nim může přistoupit kterýkoliv oprávněný spolupracovník ze zahraničních poboček.

Mým prvním úkolem po seznámení se s pracovním prostředím bylo překreslit již existující Tester univerzálních řídicích desek, který testoval životnost stykačů a vstupních kondenzátorů na řídicí desce plošných spojů. Univerzální deska a další testované součástky se používají napříč většinou přístrojů.

Na fotografii níže lze vidět na DIN liště zleva přichycený napájecí zdroj, výkonové rezistory, analogové časovače, stykače a mechanické pulzní čítače. Silové černé kabely namotané okolo pulzních čítačů představují parazitní indukčnosti kabelů zapojených v přístrojích. Dole jsou pak přišroubovány na distančních sloupcích tři řídicí desky.



Obrázek 1: Tester univerzálních řídících desek

Překreslit zapojení vodičů nebylo obtížné. Musel jsem si dát pozor na správné umístění a barvu izolace. To vše jsem několikrát zkontroloval. První verze schématu Testeru desek je v mé práci uložena jako [příloha A](#). Schéma je zde uloženo bez rohového razítka a okrajů, které neobsahují žádné užitečné informace.

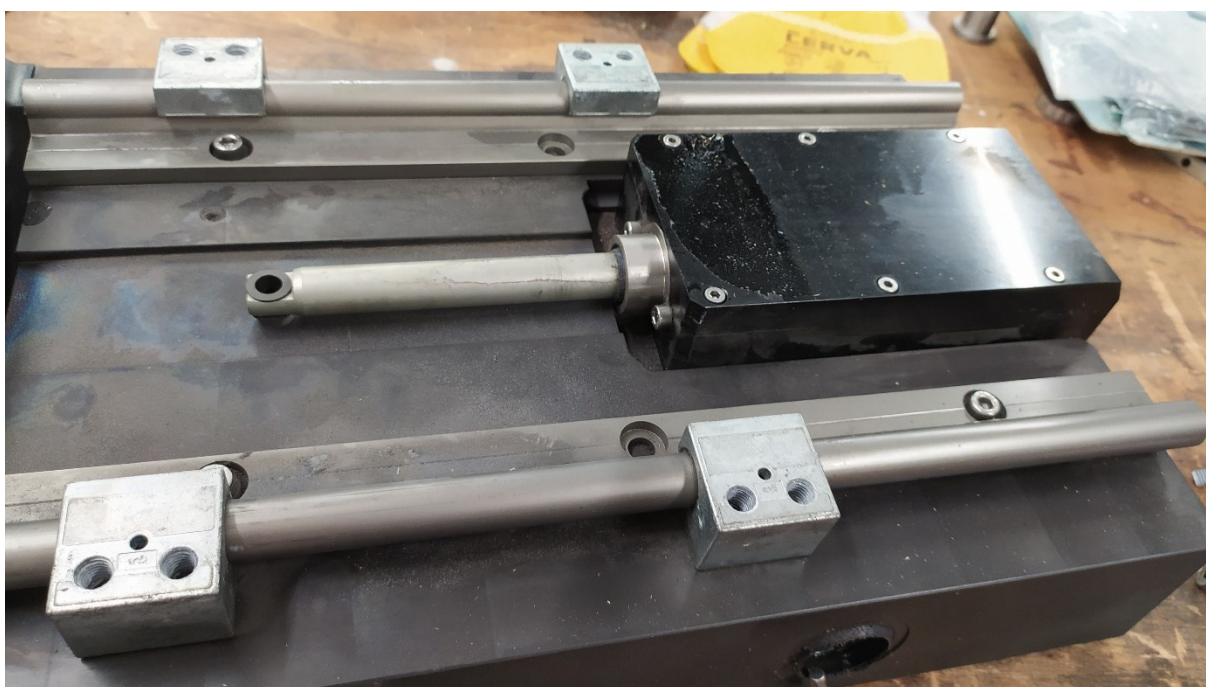
Své výsledky jsem konzultoval s HW inženýrem, který mi vytkl nedostatky. Upravil jsem tedy symboly pro analogové časovače a výkonové stykače. Změnil jsem blok se vstupním napájením a zpřehlednil vodivé cesty. Dále jsem nevěděl, že dlouhé černé kabely mají úmyslně působit jako parazitní indukčnost, a proto jsem je tak i nakonec zaznamenal do schématu pod označením L1 až L4. Finální verze je uložena jako [příloha B](#). Tato verze je nyní dostupná ve firemní databázi.

2.2 Teplotní a vlhkostní senzor

Dalším úkolem bylo vytvořit mikrokontrolér, který by pravidelně sbíral data ze tří teplotních senzorů a jednoho senzoru vlhkosti. Tyto hodnoty by pak zobrazoval na displeji a zároveň zapisoval na paměťovou kartu i s časovým razítkem, které by získal z obvodu aktuálního času.

Pro tento a v podstatě všechny další úkoly jsem vybral mikrokontrolér ATmega328P [3] postaven na elektronické vývojové platformě Arduino Uno, protože má naprosto dostatečné množství vstupně/výstupních portů, má ideální velikost pro držení v ruce při testování a velmi snadno se k jeho základní desce připojují rozšiřující moduly.

V jednom z řezacím zařízení zákazníka se poškodil vlhkostí malý krokový motor PKP245D15B-L [4], který v jedné ose posunoval kovovou desku, na které leží řezaný vzorek. Tato deska je přichycena k velké kovové podložce, v níž se nachází komora pro umístění motoru. Předpokládalo se, že problém způsobil rozdíl v teplotní kapacitě malého krokového motoru a kovového odlitku (neboli krytu motoru), který se nacházel v úzké mezeře mezi velkými kovovými částmi řezacího zařízení. Při ukončení procesu řezání mohou mít vnitřní části zařízení přes 50 °C. Velké kovové díly zůstávají teplé i dlouho po otevření krytu zařízení, ale malý krokový motor se ochladí pravděpodobně rychleji. Měřením časového průběhu chladnutí konstrukce zařízení se mělo zjistit, jestli opravdu docházelo k příliš velkému rozdílu teplot krokového motoru a jeho krytu. Na fotografii níže je vidět konstrukce pro posouvání řezací desky.



Obrázek 2: Kovová podložka řezací desky a kryt krokového motoru

První teplotní senzor byl umístěn na těle malého motoru, druhý visel ve vzduchu v krytu motoru společně s vlhkoměrem a třetí byl připevněn na stěně odlitku.

K měření tří teplot a vlhkosti jsem použil tři voděodolné senzory teploty, které umí komunikovat na sběrnici One-Wire DS18B20 [\[5\]](#) a jeden modul pro měření teploty a vlhkosti AM2302 [\[6\]](#). Pro ukládání dat a aktuální čas jsem vybral rozšiřující modul Arduino Data Logger Shield. Ten obsahuje slot pro SD kartu a obvod reálného času DS1307 s krystalem. Na tento modul jsem připojil další rozšiřující modul DFR0009 obsahující LCD displej a 5 tlačítek.

```
#include <Wire.h>
#include <SPI.h>
#include <SD.h>
#include "RTCLib.h"
RTC_DS1307 DS1307;
const int sd_CS = 10;
const int analogPin = A0;

#include <OneWire.h>
#include <DallasTemperature.h>
#define ONE_WIRE_BUS 2
OneWire oneWire(ONE_WIRE_BUS);
DallasTemperature sensors(&oneWire);

#include "DHT.h"
#define DHTPIN 3
#define DHTTYPE DHT22
DHT dht(DHTPIN, DHTTYPE);

//64bit ADRESY SENZORŮ Z DÍLNY
uint8_t MOTOR[8] = { 0x28, 0x30, 0xD7, 0xC6, 0x0B, 0x00, 0x00, 0x0D };
uint8_t VZDUCH[8] = { 0x28, 0x43, 0x2B, 0xC6, 0x0B, 0x00, 0x00, 0x4B };
uint8_t KRYT[8] = { 0x28, 0x67, 0xD1, 0xC3, 0x0B, 0x00, 0x00, 0xBB };
```

V první části kódu jsem přidal do programu knihovny pro obsluhu senzorů, čtení a zápis na SD kartu a práci s RTC obvodem. Všechny senzory umožňující komunikaci přes sběrnici One-Wire mají vlastní unikátní adresu. Adresy použitých senzorů jsou uloženy v poli osmibitové proměnné. Každé pole je pojmenováno podle pozice, kde bude teplotní senzor umístěn.

```
DeviceAddress adresa;
sensors.getAddress(adresa, 0);
for(int i = 0; i < 8; i++) {
    Serial.print("0x");
    if(adresa[i] < 0x10) Serial.print("0");
    Serial.print(adresa[i], HEX);
    if(i < 7) Serial.print(" ");
}
```


Senzory DS18B20 je možné adresovat i indexy, které jim přidělí mikroprocesor při inicializaci programu. Tuto možnost jsem ale neměl, protože teplotní senzory už byly ve stroji připevněné a bylo obtížné se k nim dostat. Proto jsem nejdřív celý program zprovoznil s jinými třemi senzory stejného typu. Až poté jsem ve výrobní hale připojil Arduino jednotlivě ke třem sensorům a příkazem *sensors.getAddress(adresa, 0)* zjistil adresy připojených sensorů. Adresy se uložily do speciální proměnné *DeviceAddress* a poté pomocí cyklu *for* vytiskli skrz sériovou komunikaci jako osm správně naformátovaných hexadecimálních čísel.

```
void loop() {  
  sensors.requestTemperatures();  
  
  float tempMOTOR = sensors.getTempC(MOTOR);  
  lcd.setCursor(0, 0);  
  lcd.print(tempMOTOR, 1);  
}
```

Funkce *loop* typu *void* je hlavní programová smyčka mikrokontroléru ATmega328P. V ní se obvykle nachází hlavní část kódu celého programu a tato smyčka se opakuje, dokud je mikrokontrolér napájen. Pomocí funkce *sensors.requestTemperatures()* přikáže mikroprocesor všem sensorům připojeným na sběrnici změřit teplotu a uložit ji ve své EEPROM. Příkazem *sensors.getTempC(adresa)* pak Arduino kontaktuje konkrétní senzor, který po sběrnici odešle naměřená data. Při tisknutí textu na LCD je třeba nejdřív nastavit sloupec a řádek, kterým bude začínat vypisovaný text. K tomu slouží funkce *lcd.setCursor(0, 0)*, kde je první vstupní hodnotou sloupec a druhou řádek umístění kurzoru. Příkaz *lcd.print()* pak vytiskne číselnou proměnnou, textový řetězec nebo speciální znak. Textem může být řetězec znaků, nebo číslo. Nikdy však obojí najednou.

```
DateTime datumCas = DS1307.now();  
String thisSecond;  
  
if (datumCas.second() < 10){  
  thisSecond = "0" + String(datumCas.second());  
}  
else {  
  thisSecond = String(datumCas.second());  
}
```

DateTime je speciální proměnná do které se příkazem *DS1307.now()* ukládá aktuální čas, jenž poskytuje RTC obvod z rozšiřujícího Data Logger Shield modulu. Další část kódu slouží k ošetření toho, aby se hodnota sekund menší než deset nezapisovala jako jednociferné číslo. ATmega328P vytvoří textový řetězec a pokud bude hodnota sekund menší než deset, přidá se do řetězce před číslo sekund nula. Pokud je číslo sekund větší, uloží se do řetězce jen samotné číslo. Stejným způsobem se budou formátovat minuty, hodiny, dny i měsíce. Časové razítko tak bude mít vždy konstantní délku, což znamená, že bude více přehledné a bude se s ním lépe pracovat.

```
File zapisDat = SD.open("mereni.txt", FILE_WRITE);
delay(10);
zapisDat.print(thisDay + ". " + thisMonth + ". " + thisYear + " - " + thisHour + ":" + thisMinute + ":" +
thisSecond);
delay(10);
zapisDat.print(" | Motor: " + String(tempMOTOR, 1) + " °C, Vzduch: " + String(tempVZDUCH, 1) + "
°C, Krypt: " + String(tempKRYT, 1));
delay(10);
float h = dht.readHumidity();
zapisDat.println(" °C, Vlhkomet: " + String(dht.readTemperature(), 1) + " °C, " + String(h, 1) + " %");
```

Poslední částí kódu teplotního a vlhkostního senzoru se týká zápisu dat na SD kartu. *File* inicializuje speciální proměnnou pro práci s SD kartou. Příkaz *SD.open* otevře textový soubor umístěný na kartě. Pokud se tam žádný takový soubor nenachází, mikroprocesor ho vytvoří. Funkce *zapisDat.print* tiskne do otevřeného textového dokumentu řetězec textu. Tisknutý text v rámci jedné funkce nemůže být příliš dlouhý, protože pokud by byl, přetekla by vyrovnávací paměť procesoru mikrokontroléru. Tato situace by způsobila, že se na SD kartu запиše pouze část textového řetězce. Zápis textu jsem proto rozdělil na tři části. Nejprve se запиše správně naformátované časové razítko, poté změřené hodnoty teplotních senzorů a na konec se zjistí vlhkost ze senzoru AM2302, která se do textového souboru vloží také. Mezi každým zápisem je nastaveno deseti milisekundové zpoždění *delay(10)*. To zajistí, že se vyprázdní vyrovnávací paměť procesoru Arduina a nedojde tak ke špatnému zápisu.

Hlavní smyčka programu mikrokontroléru se periodicky opakuje každou druhou sekundu. To zajišťuje dostatečně přesné informace o tom, co se dělo v komoře s elektromotorem při několikaminutovém měření časového průběhu chlazení zařízení po jeho vypnutí. Naměřená data se vložila do excelovské tabulky, kde se vynesly průběhy teplot a vlhkosti. Graf naměřených teplot je možné vidět v [příloze C](#). Při analýze měření byl nejdůležitější rozdíl aktuální teploty krytu a motoru.

Měřením se prokázalo, že by při běžné okolní vlhkosti a teplotě nemělo docházet ke kondenzaci vody v komoře motoru. Mechanický inženýr, který měření a jeho vyhodnocování prováděl, nyní pracuje na implementaci ventilu s filtrem, který by měl zabránit vniku okolní vlhkosti do zařízení, a to i v prostředí s vyšší vlhkostí vzduchu.

2.3 Joystick tester

V mnoha, ve firmě vyráběných zařízeních, se používá joystick pro ovládání menu a vodních trysek. Tyto joysticky se kupují od externích dodavatelů, což znamená, že je třeba ověřit plnou funkčnost každého kusu. Dostal jsem proto úkol vytvořit zařízení na bázi mikrokontroléru pro jednoduché testovací správného zapojení a funkce joysticků. Joystick je připojený k řídící desce desetižilovým plochým kabelem. Funkce jednotlivých pinů konektoru joysticku jsou v tabulce níže.

Tabulka 1: Zapojení pinů Joysticku

Pin	1	2	3	4	5	10
Funkce	GND	X axis	Y axis	Rotation	Push	+5V

Chybějící piny jsou nepřipojené. První a desátý pin slouží jako napájení joysticku. Druhý a třetí zajišťují pohyb po vodorovné a svislé ose. Čtvrtý a pátý pin určuje otočení páky a její zmáčknutí. Joysticky v některých výrobcích nemají funkci otáčení páky, ale to pro měření není problém.

Joystick je napájen přímo z Arduina. Uvnitř joysticku se nachází integrovaný obvod, který reaguje na pohyb páky změnou výstupního napětí pinů dva až pět. Toto napětí měří Arduino analogovými vstupy A1 až A5 pomocí analogově digitálního převodníku. Ten má rozlišení 1024 bitů a měří na rozsahu 0 až 5 V. Napěťové hodnoty výstupů testovaného joysticku se pohybují v rozmezí 0,5 až 4,5 V.

```
float U1;
U1 = analogRead(A1) * (5.0 / 1023.0);
```

U1 je proměnná ukládající reálné číslo. Do této 32bitové proměnné se uloží napětí měřené na druhém pinu. Funkce *analogRead(A1)* vrátí celočíselnou hodnotu změřenou analogově digitálním převodníkem z rozsahu 0 až 1023 na vstupu A1. Toto číslo je vynásobeno napěťovým rozsahem a poděleno maximální možnou hodnotou výstupu ADC. Napětí 5 V na vstupu tak bude odpovídat přesně hodnotě 5 uložené v proměnné *U1*.

```
lcd.setCursor(0,0);  
lcd.print("I1=");  
lcd.print(U1,1);  
lcd.print("V");  
  
lcd.setCursor(9,0);  
lcd.print("I2=");  
lcd.print(U2,1);  
lcd.print("V");  
.  
.  
delay(100);  
} //konec hlavní smyčky
```

Měřené hodnoty jsem na displej umístil tak, aby se mohly všechny čtyři zobrazit najednou. Napětí se zobrazují jako čísla s jednou cifrou za desetinnou čárkou. V textu vypisovaném na LCD chybí mezery, ale ty pro tuto aplikaci nejsou podstatné. Krom toho by se korektně formátovaný text na displej ani nevešel. Hlavní programová smyčka, a tedy i obnova informací na LCD, se opakuje asi desetkrát za sekundu. To způsobuje příkaz `delay()`, jehož jediným parametrem je délka čekání procesoru v milisekundách. Tento příkaz není vhodný pro složitější programy, protože při jeho vykonávání procesor neprovádí žádnou jinou činnost. V tomto případě to ale nevadí.



Obrázek 3: Funkční Joystick tester

Fyzickou podobu testeru jsem vyrobil dle schématu uloženého jako [příloha D](#). Ve schématu lze vidět, že 4 piny vyvedené z joysticku se nevyužívají. Piny pro určení polohy páky jsou přizemněny skrz rezistory s velkým odporem. To zajišťuje měření nulového napětí při odpojení pinu.

Napětí výstupu joysticku pro základní polohu svislé a vodorovné osy je přesně 2,5 V. Při pohybu vlevo a dolů se napětí snižuje, a naopak při pohybu vpravo a nahoru zvyšuje. Pro senzor otočení páky platí taktéž, že je v základní poloze napětí senzoru 2,5 V. Při otáčení proti směru hodinových ručiček se napětí snižuje, a při otáčení po směru hodinových ručiček se zvyšuje. Páka se dá stlačit pouze dolů a napětí pátého pinu je v klidovém stavu přibližně 0,5 V.

Tento Joystick tester se ve výrobní hale pravidelně používá pro testování joysticků umístěných ve vyráběných zařízeních.

2.4 Foil tester

Dalším úkolem bylo vytvořit testovací zařízení na bázi mikrokontroléru, které bude ověřovat správné zapojení tlačítek v nalepujících se čelních panelech vyráběných přístrojů. Těm se v rámci firmy říká fólie (anglicky foil). Fólie se, stejně jako joysticky, kupují od externích dodavatelů, a proto je nutné ověřit funkčnost každého kusu.

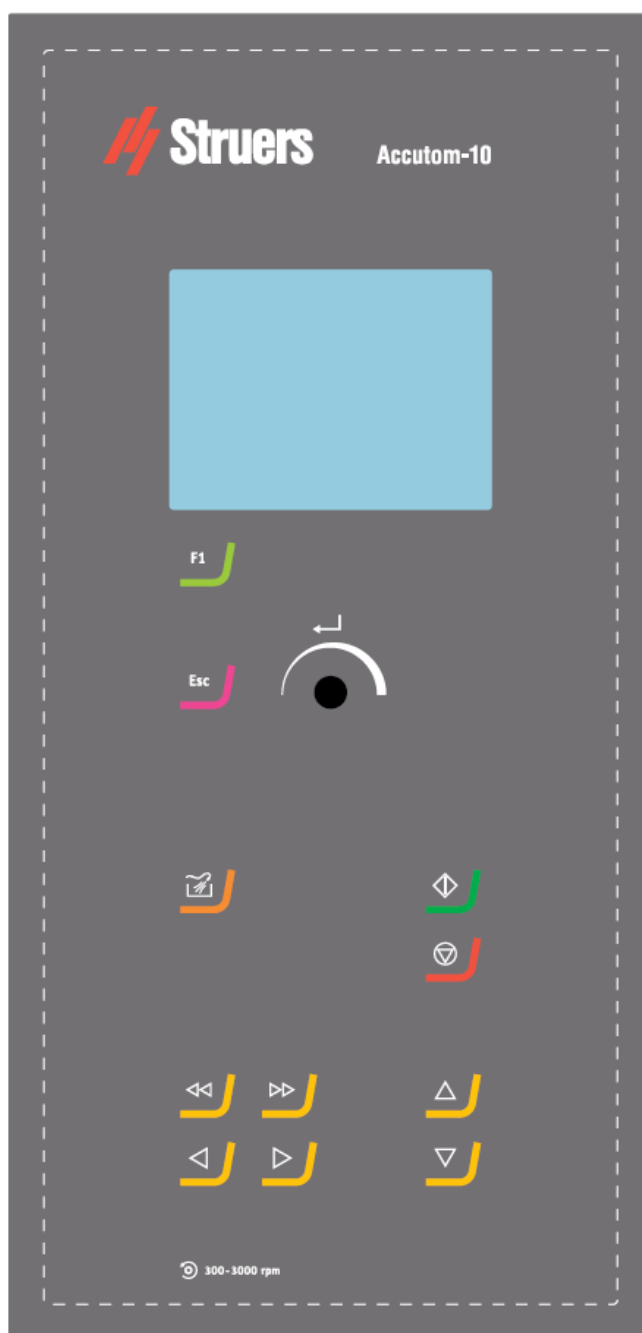
Tabulka 2: Příklad uspořádání tlačítek fólie

Key	Function	Colour	PIN 1	PIN 2	PIN 3	PIN 4	PIN 5	PIN 6	PIN 7	PIN 8	PIN 9	PIN 10
1	F1	Limegreen	X						X			
5	Escape	Pink		X					X			
9	Flushing gun	Orange			X				X			
11	Start	Green			X						X	
12	Stop	Red			X							X
13	Move X-axis to the left fast	Yellow				X			X			
14	Move X-axis to the right fast	Yellow				X				X		
16	Move Y-axis to the back	Yellow				X						X
17	Move X-axis to the left slowly	Yellow					X		X			
18	Move X-axis to the right slowly	Yellow					X			X		
20	Move Y-axis to the front	Yellow					X					X
	Jumper*							X		X		

*Make a short circuit in the foil which corresponds to a permanent activated key

Všechny fólie se k hlavní desce připojují desetižilovým plochým kabelem. Ve fólii se nachází matice tlačítek zapojených například dle tabulky na předchozí straně. Zmáčknutím konkrétního tlačítka se vodivě spojí dva vodiče označené písmenem X. Fólie pro jednotlivé přístroje se pak identifikují permanentním vodivým spojením dvou vybraných vodičů, čímž tím je zajištěno, že se jednou fólií nedá ovládat přístroj jiného typu.

Pro vytvoření fyzického testeru jsem potřeboval platformu Arduino Uno, rozšiřující modul s LCD, zásuvku pro desetižilový plochý kabel a deset vodičů.



Obrázek 4: Fólie používaná pro ovládání řezacího stroje

2.4.1 První verze Foil testeru

První verze řešení tohoto úkolu tedy spočívala v cyklicky se opakující smyčce programu, který prvně identifikoval připojenou fólii pomocí identifikačního klíče a dále zjišťoval zmáčknutí jednotlivých tlačítek. Informaci o testované fólii a zmáčknutém tlačítku poté zobrazoval na displeji.

```
#include <LiquidCrystal.h>
LiquidCrystal lcd(8,9,4,5,6,7);

byte x1 = 0;           //D0 x1
byte x2 = 1;           //D1 x2
byte x3 = 2;           //D2 x3
byte x4 = 3;           //D3 x4
byte x5 = 11;          //D11 x5
byte x6 = 12;          //D12 x6

byte x7 = A5, x7_state = 0; //A5 x7
byte x8 = A2, x8_state = 0; //A2 x8
byte x9 = A3, x9_state = 0; //A3 x9
byte x10 = A4, x10_state = 0; //A4 x10
```

Pro tento program mi stačila pouze knihovna obsluhující LCD. Dále jsem si v hlavičce programu definoval proměnné, které reprezentují jednotlivé piny desetižilového konektoru a proměnné, do kterých bude mikrokontrolér zapisovat digitální napěťovou úroveň na těchto pinech.

```
byte foil = 0;
digitalWrite(x5, HIGH);
x7_state = digitalRead(x7);
x8_state = digitalRead(x8);

lcd.setCursor(0,0);
lcd.print("-----");

if (x7_state == HIGH) {
  lcd.setCursor(0,0);
  lcd.print("Foil: Secotom-20");
  foil = 1;
}
if (x8_state == HIGH) {
  lcd.setCursor(0,0);
  lcd.print("Foil: Secotom-60");
  foil = 2;
}
```

Osmibitová celočíselná proměnná *foil* slouží pro uložení typu fólie. Identifikace fólie se provádí tak, že se zvýší napětí na jednom z pinů fólie a měří se další piny fólie. Pokud dojde ke změně vyšší logické úrovně na konkrétním pinu, je zřejmé, že došlo k vodivému propojení těchto dvou pinů. Na displej Arduina se vypíše typ zařízení, kterému odpovídají zjištěné identifikační piny a do proměnné *foil* se uloží číslo fólie. K vypisování šestnácti pomlček na prvním řádku displeje dochází proto, aby po odpojení fólie nezůstala informace o jejím připojení napsaná na displeji. Pokud je k testeru připojena fólie, jsou pomlčky přepsány tak rychle, že si toho lidský mozek nemůže všimnout.

```
digitalWrite(x1,HIGH);
x9_state = digitalRead(x9);
x10_state = digitalRead(x10);

lcd.setCursor(0,1);
lcd.print("          ");

if (x9_state == HIGH && ( x==1 || x==2 ) ) {
  lcd.setCursor(0,1);
  lcd.print("Button: WATER  ");
}
if (x10_state == HIGH && ( x==1 || x==2 ) ) {
  lcd.setCursor(0,1);
  lcd.print("Button: ESC   ");
}
```

Při testování fólií se očekává zmáčknutí maximálně jednoho tlačítka v jeden okamžik. V principu je identifikace zmáčknutého tlačítka stejný proces jako identifikace připojené fólie. Navíc je zde podmínka připojení odpovídající fólie. Důvodem je to, že některé typy fólií mají stejnou kombinaci spojených pinů pro tlačítka s jiným označením. Například, pro fólii 1 může znamenat určitá kombinace spojení pinů začátek řezání a pro fólii 3 ta stejná kombinace může znamenat spuštění vodních trysek. Označení zmáčknutého tlačítka je pak vypsáno na druhém řádku LCD. I v této části kódu dochází k pravidelnému přepisování displeje proto, aby se již nezmáčknuté tlačítko nezobrazovalo.

Na konci kódu se nachází příkaz *delay(100)*. Ten způsobí, že maximální zpoždění reakce mikrokontroléru na připojení fólie nebo zmáčknutí tlačítka bude desetina sekundy. Toto zpoždění je zcela přijatelné v rámci testování tlačítek. Zpoždění je zde nutné pro přepisování displeje. Kdyby nedocházelo ke zpoždění, byl by displej půlku času v základním stavu a druhou půlku času by případně zobrazoval připojenou fólii a zmáčknuté tlačítko. To by se projevilo velmi špatnou čitelností informací na displeji.

2.4.2 Finální verze Foil testeru

Po úspěšném otestování a nasazení první varianty Foil testeru, který testoval fólie nejvíce vyráběných zařízení, přišel požadavek na možnost otestovat všechny fólie používající desetižilový plochý konektor. Těch je dohromady osm. Dalším požadavkem bylo, aby si mikrokontrolér pamatoval poslední vybranou fólii a tu při dalším spuštění automaticky vybral.

Fyzická část testeru se pro tuto verzi musela změnit. Přibyly dva rezistory regulující proud svítícími diodami, které jsou součástí fólie jednoho z vyráběných zařízení. Protože tato fólie zároveň postrádá identifikační piny, zvolil jsem variantu vytvořit tester s možností zvolení testované fólie. Vybrat si fólii k testování je možné v jednoduchém menu pomocí tlačítek na modulu s displejem.

```
byte msize = 9;           //menu size
int mpos = 0;             //menu position
String menu[] = {
  "Foil unselected ",    //0
  "Foil: Accu-10  ",     //1
  .
  .
  "Foil: Lavamin  ",     //8
};
```

Číslo proměnné *msize* odpovídá velikosti menu a proměnná *mpos* určuje aktuální pozici uživatele v menu. Vypsáný kód dále obsahuje definici devítičlenného pole s řetězcí textu, který se bude vypisovat na LCD při výběru fólie z menu.

```
byte lcd_key = 0;
byte change;

#define btnRIGHT 0    //vpravo
#define btnUP 1       //nahoru
#define btnDOWN 2     //dolů
#define btnLEFT 3     //vlevo
#define btnSELECT 4   //zvolit
#define btnNONE 5
```

Osmibitová proměnná *lcd_key* se v programu používá pro určení právě zmáčknutého tlačítka LCD modulu. Proměnná *change* stejného typu poskytuje informaci o tom, jestli již bylo v předchozím průchodu hlavním programovým cyklem zmáčknuto stejné tlačítko. Tato informace zabrání dalšímu posouvání se v menu při každém průchodu hlavním programovým cyklem. Příkaz *#define* vytvoří konstantu daného jména, která bude obsahovat zadanou hodnotu. Arduino LCD modul obsahuje pět tlačítek. Pro orientaci v menu stačí tři tlačítka, tedy *vlevo*, *vpravo* a *zvolit*. Tlačítko *nahoru* má funkci návratu na zvolenou fólii a tlačítko *dolů* vybere žádnou fólii.

```
int read_LCD_buttons() {  
int adc = analogRead(0);  
if (adc > 1000) return btnNONE;  
if (adc < 50) return btnRIGHT;  
if (adc < 250) return btnUP;  
if (adc < 450) return btnDOWN;  
if (adc < 650) return btnLEFT;  
if (adc < 850) return btnSELECT; }
```

Funkce *read_LCD_buttons* vrátí celočíselnou hodnotu odpovídající jednomu z tlačítek modulu displeje. Do proměnné *adc* je uloženo napětí na analogovém vstupu A0. Modul displeje používá analogový vstup A0 jako výstup napěťového děliče. Jednotlivá tlačítka propojují vstup A0 se zemí skrz rezistory s různými hodnotami odporu. To způsobuje různá výstupní napětí pro všechna tlačítka.

```
void loop() {  
  lcd_key = read_LCD_buttons();  
  switch (lcd_key) {  
    case btnRIGHT: {  
      lcd.setCursor(0,0);  
      if(change == 0)  
        { change = 1;  
          mpos++;  
          if ( mpos > ( msize - 1 ) ) { mpos = 0; };  
          lcd.print(menu[mpos]);  
        } break; }  
  }
```

Při průchodu každého cyklu hlavní programové smyčky se zjišťuje právě zmáčknuté tlačítko. V případě zmáčknutí tlačítka *vpravo* se zjistí, jestli již předchozí cyklus nebylo tlačítko zmáčknuťo. Pokud ne, změní se pozice menu o jednu pozici doprava. Pokud se menu dostalo za poslední položku, vrátí se zpět na první pozici. Před ukončením vícecestného větvení *switch* se na displej vypíše text vybrané fólie.

```
case btnSELECT: {  
  foil = mpos;  
  if(change == 0)  
    { change = 1;  
      EEPROM.update(0,foil);  
    } break; }
```

Zmáčknutím tlačítka *zvolit* se potvrdí aktuální výběr fólie. Číslo zvolené fólie se zároveň uloží pomocí příkazu *EEPROM.update* jako osmibitová proměnná do EEPROM s kapacitou 1024 bajtů. Příkaz *EEPROM.update* je výhodný v tom, že se hodnota v buňce paměti přepíše pouze pokud by v ní došlo ke změně. Tím se může razantně zvýšit její životnost.

```
switch (foil) {  
  case 0: { //Foil unselected  
    lcd.setCursor(0,1);  
    lcd.print("      ");  
    break; }  
  
  case 1: { //Foil: Accu-10  
    pinMode(x4, INPUT);  
    pinMode(x5, INPUT);  
    pinMode(x6, OUTPUT);  
    digitalWrite(x6,HIGH);  
    x8_state = digitalRead(x8);  
    digitalWrite(x6,LOW);  
    pinMode(x6, INPUT);  
    pinMode(x4, OUTPUT);  
    pinMode(x5, OUTPUT);  
    if (x8_state == HIGH){  
      accu_btn ();  
    }  
    else {  
      lcd.setCursor(0,1);  
      lcd.print("not connected ");  
    } break; }  
}
```

V tomto větvení, které se opakuje každý programový cyklus, se na základě zvolené fólie provádí ověření fólie a zjištění právě zmáčknutého tlačítka. Pokud nebyla zvolena žádná fólie, program vypíše na LCD prázdný druhý řádek. Pokud byla zvolena například fólie 1, mikrokontrolér ověří, že je fólie skutečně připojena. Nejprve je třeba nastavit dvěma nesouvisejícím pinům na vysokou vstupní impedanci. To je důležité kvůli případu, kdy by již bylo zmáčknuuto tlačítko, které propojuje jeden z těchto vstupů s pinem x8. V takovém případě by totiž došlo ke zkratu, který by vedl od výstupu x6 přes pin x8 až do vstupu x5 nebo x4. Tuto situaci je možno lépe pochopit z [tabulky 2](#).

Po ověření, že došlo k připojení správné fólie, spustí program funkci *accu_btn ()*. V ní se nachází kód totožný s kódem v předchozí podkapitole na straně 20. Dochází zde ke zjišťování propojených pinů, které odpovídají tlačítkům na fólii. Pro každou řadu fólií existují různé kombinace propojených pinů pro jinak označená tlačítka. Proto má každá řada vlastní funkci pro zjišťování propojených pinů.

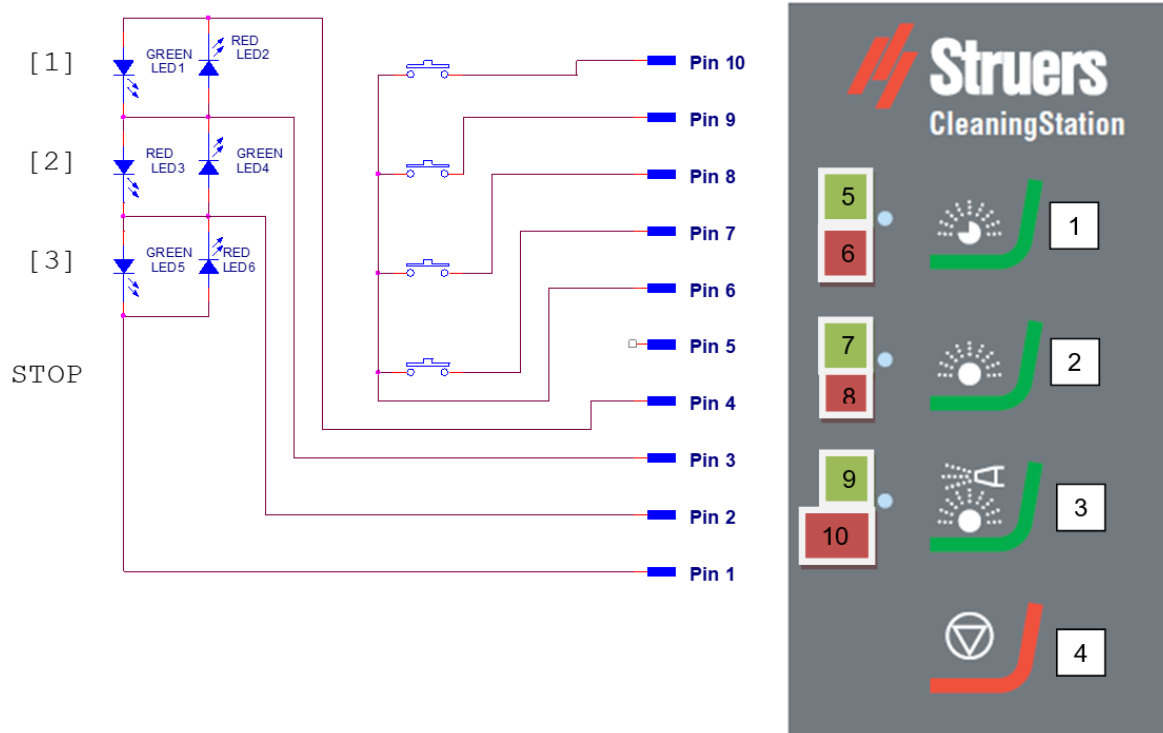
V případě, že fólie není připojena, nebo je připojena špatně, se na LCD vypíše chybová hláška upozorňující na tento problém. Tímto větvením v podstatě končí hlavní programová smyčka.

```

if(i < 5) {
  digitalWrite(x1,LOW);
  digitalWrite(x3,LOW);
  digitalWrite(x2,HIGH);
  digitalWrite(x4,HIGH);
  i++;
}
else {
  digitalWrite(x2,LOW);
  digitalWrite(x4,LOW);
  digitalWrite(x1,HIGH);
  digitalWrite(x3,HIGH);
  i++;
}
if(i >= 10) {
  i=0; }

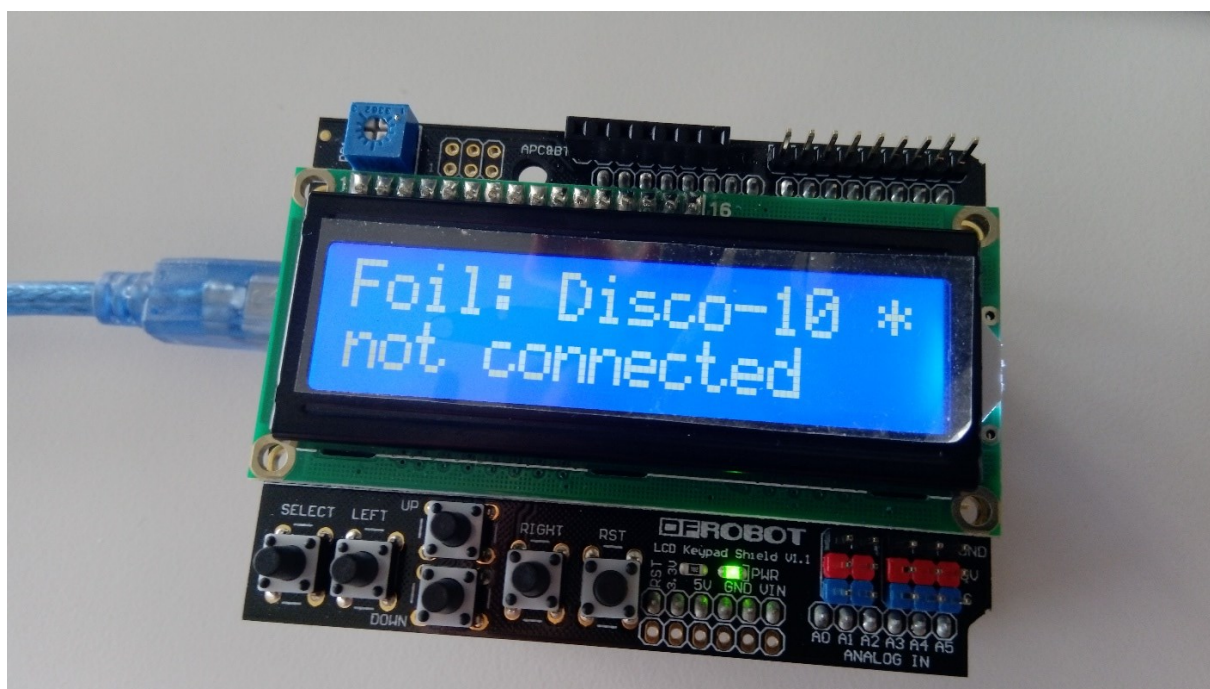
```

Jedna z fólií obsahuje krom tlačítek i svítivé diody, které nahrazují displej. Jedná se o čelní panel ultrazvukového čističe, který nepotřebuje ke své obsluze menu s displejem. Šest diod je zapojeno dle obrázku 5 v maticovém antiparalelním zapojení. Program mikrokontroléru bude prvních pět průběhů hlavního cyklu zapínat všechny zelené LED a dalších pět průběhů všechny červené. Jeden hlavní programový cyklus trvá minimálně 100 milisekund, takže se LED přepínají přibližně dvakrát za sekundu.



Obrázek 5: Schéma zapojení a obrázek fólie Lavamin

Foil tester jsem zhotovil dle schématu v [příloze E](#). Samotný tester je možné vidět na fotografii níže. Foil tester se používá ve firmě pravidelně pro testování všech vyrobených a nalepených fólií. V průběhu času přišel dokonce požadavek na vyrobení dalšího kusu. Při této příležitosti jsem požádal HW mechanika, aby mi namodeloval a vytiskl dvě plastové krabičky. V plastové krabičce se Arduino lépe drží a je mnohem odolnější, což je pro prostředí výroby velmi podstatné. Finální výrobek je zachycen na fotografii v *obrázku 6* níže na této stránce.



Obrázek 6: Funkční Foil tester



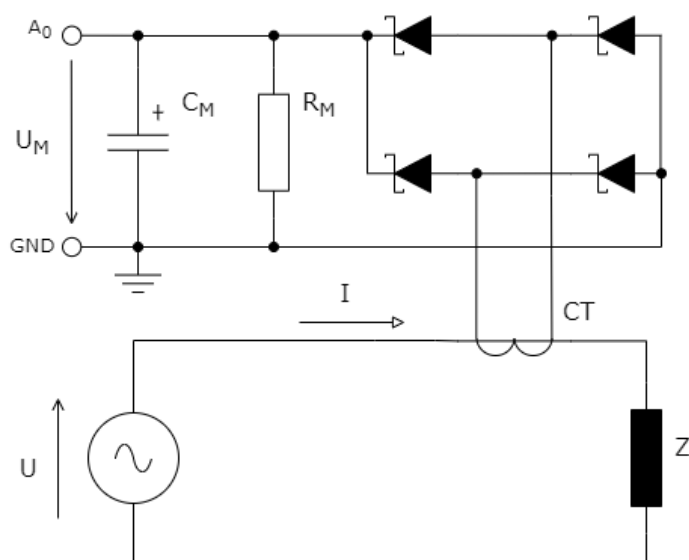
Obrázek 7: Foil tester ve vytisknuté plastové krabičce

2.5 LED indikátor výkonu

Mým aktuálním úkolem je navrhnout a sestavit indikátor příkonu umístěný v krabici s napájecím vstupem, přepínačem, patnácti LED a zásuvkou, který bude indikovat elektrický výkon odebíraný ze zásuvky rozsvícením jedné z LED. Indikátor výkonu se bude používat pro rozpoznání správného nastavení trimru. Ten se nachází na řídicí desce a jeho úkolem je upravovat časování spínání výkonových prvků. Tyto polovodičové spínače pak napájí ultrazvukový měnič v ultrazvukové čističce. Rozsah měřeného výkonu bude 80 až 140 W. Každá LED tedy bude indikovat rozsah výkonu 4 W. LED jsou obdélníkového tvaru a jsou seskládány zleva doprava v pořadí 4 červené, 2 oranžové, 3 zelené, 2 oranžové a opět 4 červené. Pokud se rozsvítí jedna ze tří zelených diod, operátor ví, že má správně nastavený trimr a ultrazvukový měnič je napájen výkonem, na který je dimenzován.

K měření procházejícího elektrického proudu budu používat neinvazivní proudový transformátor AX-0500 [7]. Proudový transformátor se v elektrickém obvodu chová jako proudový zdroj. Je proto vhodné ho zatížit co nejmenším odporem. Menší odpor ale způsobí omezení napěťového rozsahu měření. Výrobci CT proto uvádějí vhodnou hodnotu zatěžovacího odporu v katalogových listech. Výstupní proud použitého proudového transformátoru je v ideálním stavu pětsetkrát menší než proud měřeného obvodu.

Pro tuto aplikaci využiji vývojovou desku Arduino Uno. Schéma zapojení mikrokontroléru je na obrázku níže. Proud proudového transformátoru je usměrněn můstkovým usměrňovačem složeným ze Schottkyho diod, které jsou charakteristické nízkým napěťovým úbytkem v propustném směru. Usměrněný proud vytvoří úbytek napětí na zatěžovacím odporu. Měřené napětí je vyhlazeno elektrolytickým kondenzátorem a je přímo úměrně závislé na proudu měřeného obvodu.

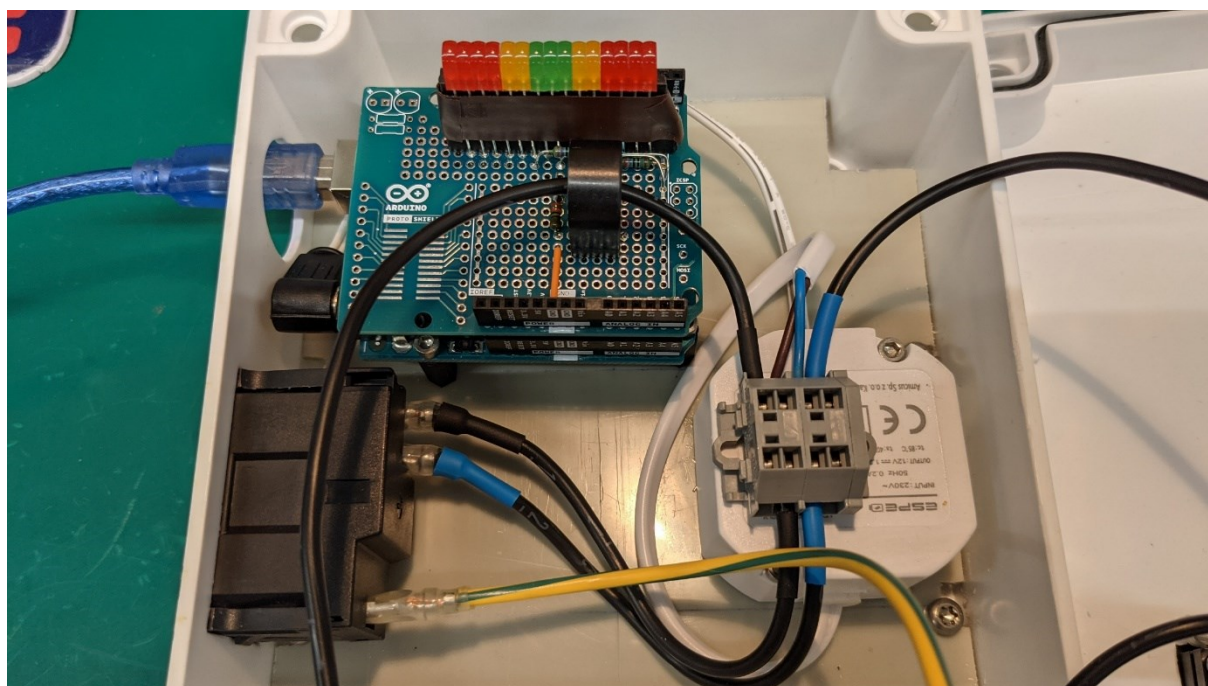


Obrázek 8: Schéma zapojení mikrokontroléru pro měření střídavého proudu

K měření skutečného elektrického výkonu střídavého napětí je velikost procházející proudu nedostatečná informace. Proto budu mít při prvním testování celého indikátoru výkonu do obvodu zapojený i digitální wattmetr. Při nastavování trimru řídicí desky budu porovnávat hodnotu změřeného výkonu wattmetru a napětí, které změřil mikrokontrolér na zatěžovacím odporu transformátoru. Teoreticky by mezi oběma změřenými hodnotami měla být lineární závislost. Ze zmíněného měření pak získám výpočtem konstantu, kterou bude mikrokontrolér násobit změřené napětí na zatěžovacím odporu proudového transformátoru.

Tato metoda měření ale nebude počítat s kolísáním síťového napětí. Proto k mikrokontroléru připojím ještě modul pro měření síťového napětí s transformátorem napětí. MCU tak bude měřit průběh střídavého napětí, ze kterého získá jeho amplitudu. Hodnota amplitudy síťového napětí se pak bude porovnávat se uloženou hodnotou amplitudy síťového napětí z prvního měření. Podíl těchto dvou hodnot napětí se vloží do rovnice výpočtu odebíraného výkonu společně se změřeným proudem a vypočítaným koeficientem a výsledkem rovnice bude odebíraný výkon ze zásuvky.

Mikrokontrolérem vypočítaná hodnota výkonu pak bude lineární rovnicí převedena na číslo LED, která se má rozsvítit. Pokud bude vypočtený výkon mimo rozsah, začne blikat krajní levá nebo pravá červená LED. Na fotografii níže je vidět aktuální stav LED indikátoru příkonu. Doufám, že i tento zadaný úkol brzy úspěšně dokončím a že bude užitečný při výrobě ultrazvukových čističek.



Obrázek 9: Nedokončený LED indikátor výkonu

3 Teoretické a praktické znalosti a dovednosti získané v průběhu studia uplatněné během odborné praxe

Během odborné praxe jsem nejvíce uplatnil znalost a dovednost základů elektroniky, kreslení elektrických schémat, pájení a programování. Mnohým z těchto dovedností jsem se naučil již na střední škole. K získání samotné práce mi taky pomohlo absolvování předmětu *Soft Skills*. Paní Mgr. Martina Schneiderová, Ph.D., která tento předmět vedla, nám pomohla vypracovat náš životopis, sepsat motivační dopis a vyzkoušet si i přijímací pohovor.

Pro první samostatný úkol ve firmě jsem potřeboval především znalost programu OrCAD. Důležité bylo umět najít v knihovnách značku používaných elektrických součástek a orientovat se v prostředí programu. Další úkoly vyžadovaly schopnost orientovat se v katalogových listech, mít přehled o funkcích a vlastnostech různých výkonových elektrických součástek a umět pracovat s multimetrem, osciloskopem a napájecím zdrojem.

Můj aktuální úkol ve firmě, tedy indikátor výkonu, vyžaduje znalost měření elektrických veličin proudu a napětí. Především jde o princip fungování neinvazivního proudového transformátoru a jeho správného použití. Jsem si jist, že s prohloubením mých znalostí v rámci studia na vysoké škole se zvýší i odborná náročnost úkolů, které budu dostávat ve firmě. Především se bude jednat o znalosti z oblasti řízení elektromotorů, bezpečnosti v elektrotechnice, elektrotechnických předpisech a elektromagnetickou kompatibilitu.

4 Znalosti či dovednosti scházející studentovi v průběhu odborné praxe

Má první neznalost se týkala celého systému fungování databáze dokumentů a schémat firmy. Každý typ dokumentu má své předčísí, pak následuje číslo projektu a poslední trojčíslí je označení série dokumentu. Toto číslo se pak doplňuje do rohového razítka společně s označením autora a osoby, která dokument schválila. S tímto a v podstatě kterýmkoliv dalším problémem mi vždy rád pomohl HW inženýr.

Problémem při vyrábění testovacích zařízení bylo to, že jsem nikdy předtím nepracoval s vývojovou deskou Arduino Uno. Naštěstí, práce ve vývojovém prostředí Arduino IDE a programovací jazyk C++ jsou snadné na naučení, pokud již člověk předtím programoval v jazyku C. Další dovedností, která mi scházela byla práce s instalačními krabicemi. Bylo potřeba se naučit správně měřit a označovat, vrtat přesné díry, pilovat, vyřezávat části plastů a podobné kutilské činnosti.

5 Dosažené výsledky v průběhu odborné praxe a její celkové zhodnocení

V rámci mé individuální odborné praxe jsem odpracoval ve firmě Roper Engineering s.r.o. na pozici HW Trainee celkem 56 dní. Po celou tuto dobu jsem se dozvídal především od hlavního HW inženýra o znalostech aktuálních technologií, ale i povaze, kterou musí člověk mít, pokud chce pracovat jako odborník v této oblasti. Dozvěděl jsem se například o požadavcích na elektromotory, spolehlivosti některých řídicích desek od zahraničních dovozců, bezpečnosti na pracovišti a HW práci inženýra. Protože jsem nikdy předtím v technologické firmě nepracoval, zjistil jsem, že se takové firmy obvykle dělí na několik sekcí, které jsou odpovědné za různé části vývojového, výrobního a distribučního procesu.

Jsem rád, že jsem po dobu mé praxe pracoval v této firmě a že jsem se setkal s odborníky nejen z mého, ale i dalších technologických oborů. Všichni tito lidé mi byli nápomocní a ochotně se podělili o své znalosti a zkušenosti. Firmě jsem během své praxe přispěl několika dokončenými schématy, výrobou testerů na bázi mikrokontrérů a HW prací jako třeba pájením na PCB, detekováním a opravováním problémů s elektronikou, výrobou kabelů a konektorů a podobně.

Myslím si, že bakalářská práce formou individuální odborné praxe mi byla prospěšná a zhodnotil bych ji ze své strany pozitivně. Praxi v oboru během studia bych doporučil i dalším studentům, kteří chtějí prohloubit své znalosti a dovednosti.

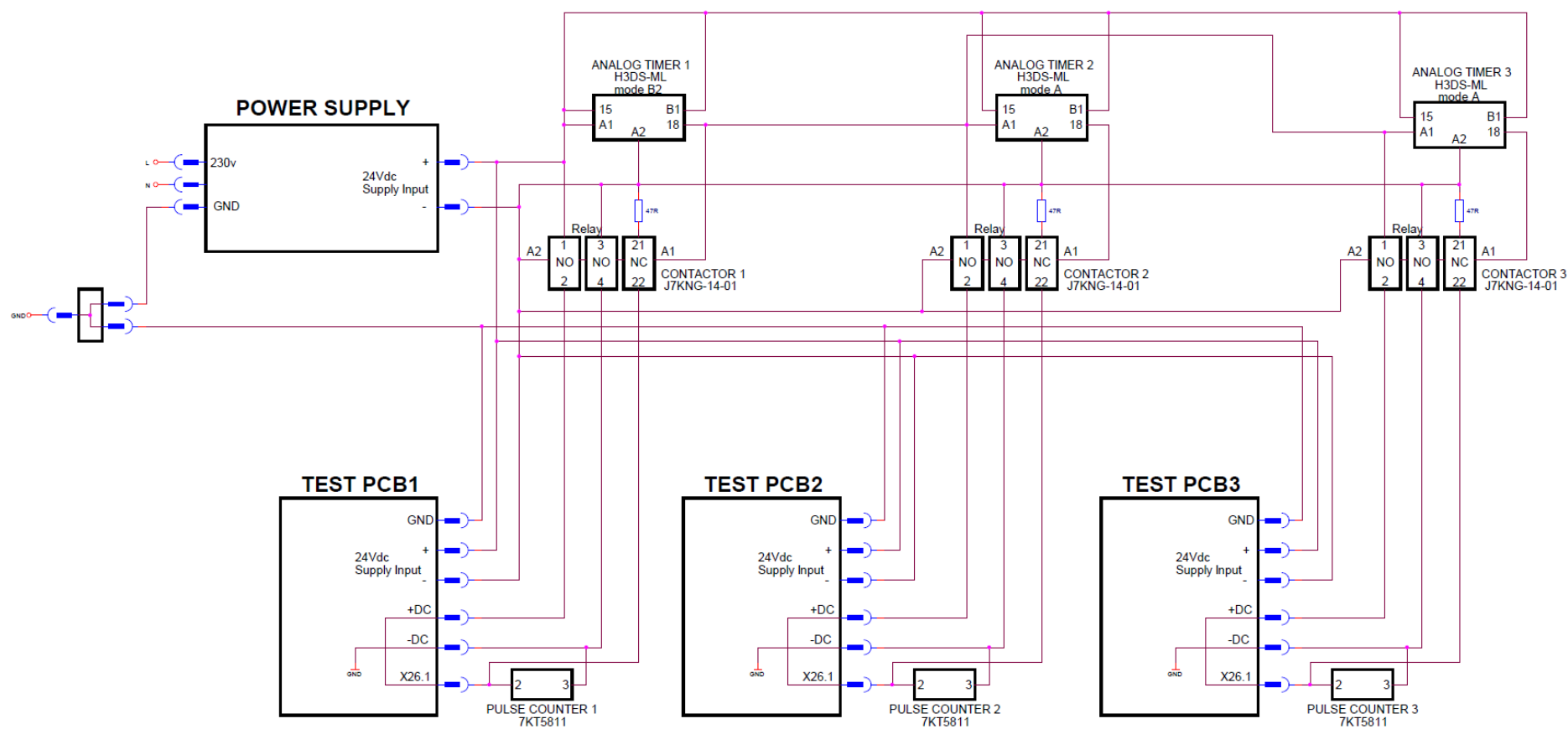
Použitá literatura

- [1] Roper Engineering [online]. Ostrava, [2021] [cit. 2021-04-28]. Dostupné z: <https://roper.cz/>
- [2] Struers: Products [online]. Ballerup (Dánsko), ©2021 [cit. 2021-04-28]. Dostupné z: <https://www.struers.com/en/Products>
- [3] Atmel Corporation [online katalogový list]. ATmega328P. ©2015 [cit. 2021-04-28]. Dostupné z: http://ww1.microchip.com/downloads/en/DeviceDoc/Atmel-7810-Automotive-Microcontrollers-ATmega328P_Datasheet.pdf
- [4] Oriental Motor U.S.A. Corp. [online katalogový list]. High-Torque Stepper Motor PKP Series. ©2016 [cit. 2021-04-28]. Dostupné z: <https://www.orientalmotor.com/products/pdfs/2015-2016/A/PKP-Series-Full-Brochure.pdf>
- [5] Maxim Integrated Products, Inc [online katalogový list]. DS18B20. ©2019 [cit. 2021-04-28]. Dostupné z: <https://datasheets.maximintegrated.com/en/ds/DS18B20.pdf>
- [6] Aosong Electronics Co.,Ltd [online katalogový list]. DHT22 (AM2302). ©2016 [cit. 2021-04-28]. Dostupné z: <https://www.sparkfun.com/datasheets/Sensors/Temperature/DHT22.pdf>
- [7] The Talema Group [online katalogový list]. AX Series : High Output Voltage Current Transformer. 2018 [cit. 2021-04-28]. Dostupné z: <https://talema.com/wp-content/uploads/datasheets/AX.pdf>

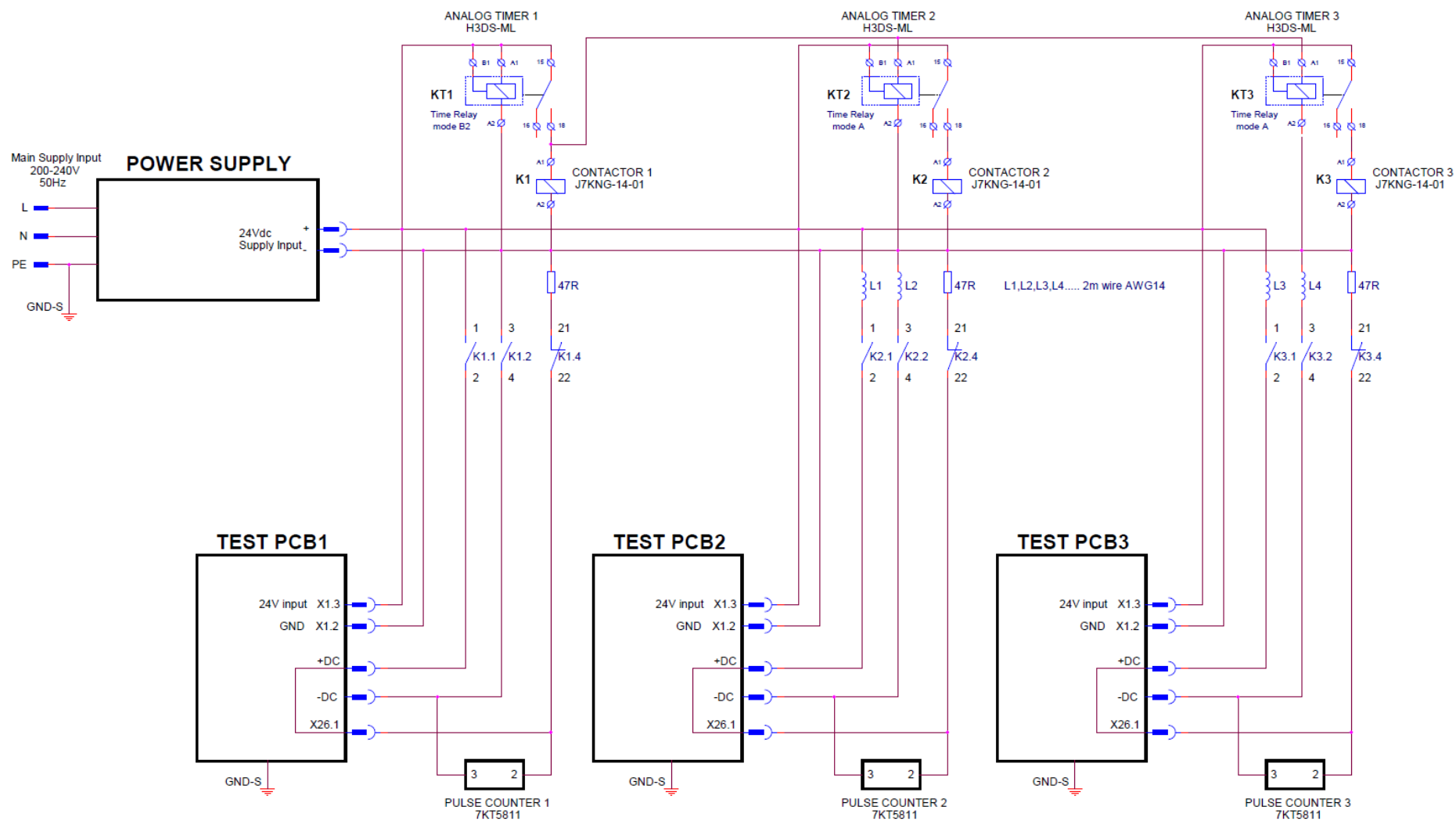
Seznam příloh

Příloha A:	První schéma zapojení Testeru univerzálních řídících desek	I
Příloha B:	Finální schéma zapojení Testeru univerzálních řídících desek	II
Příloha C:	Graf průběhu teplot odlitku, vzduchu v komoře a motoru	III
Příloha D:	Schéma zapojení Joystick testeru	IV
Příloha E:	Schéma zapojení Foil testeru	V

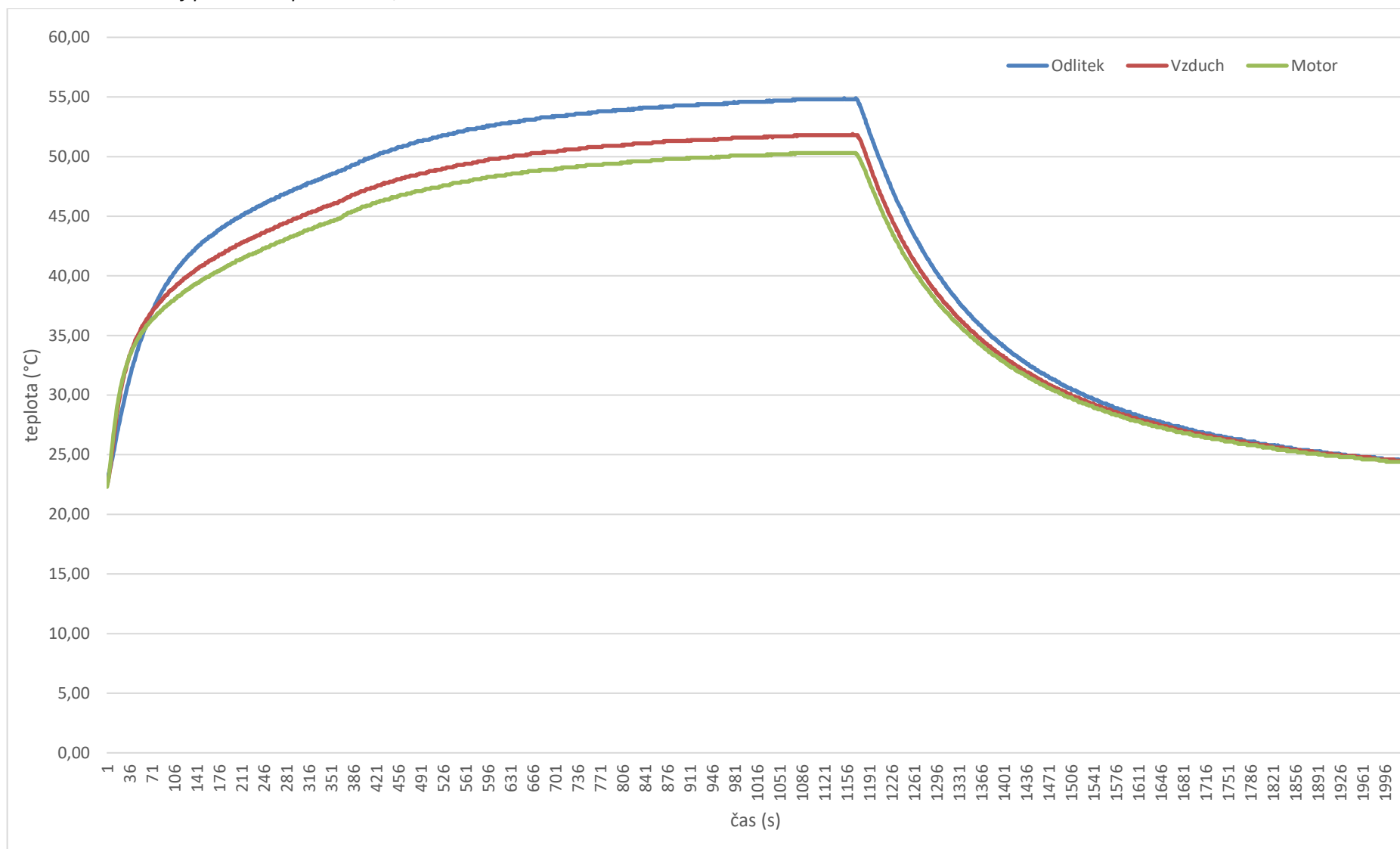
Příloha A: První schéma zapojení Testeru univerzálních řídicích desek



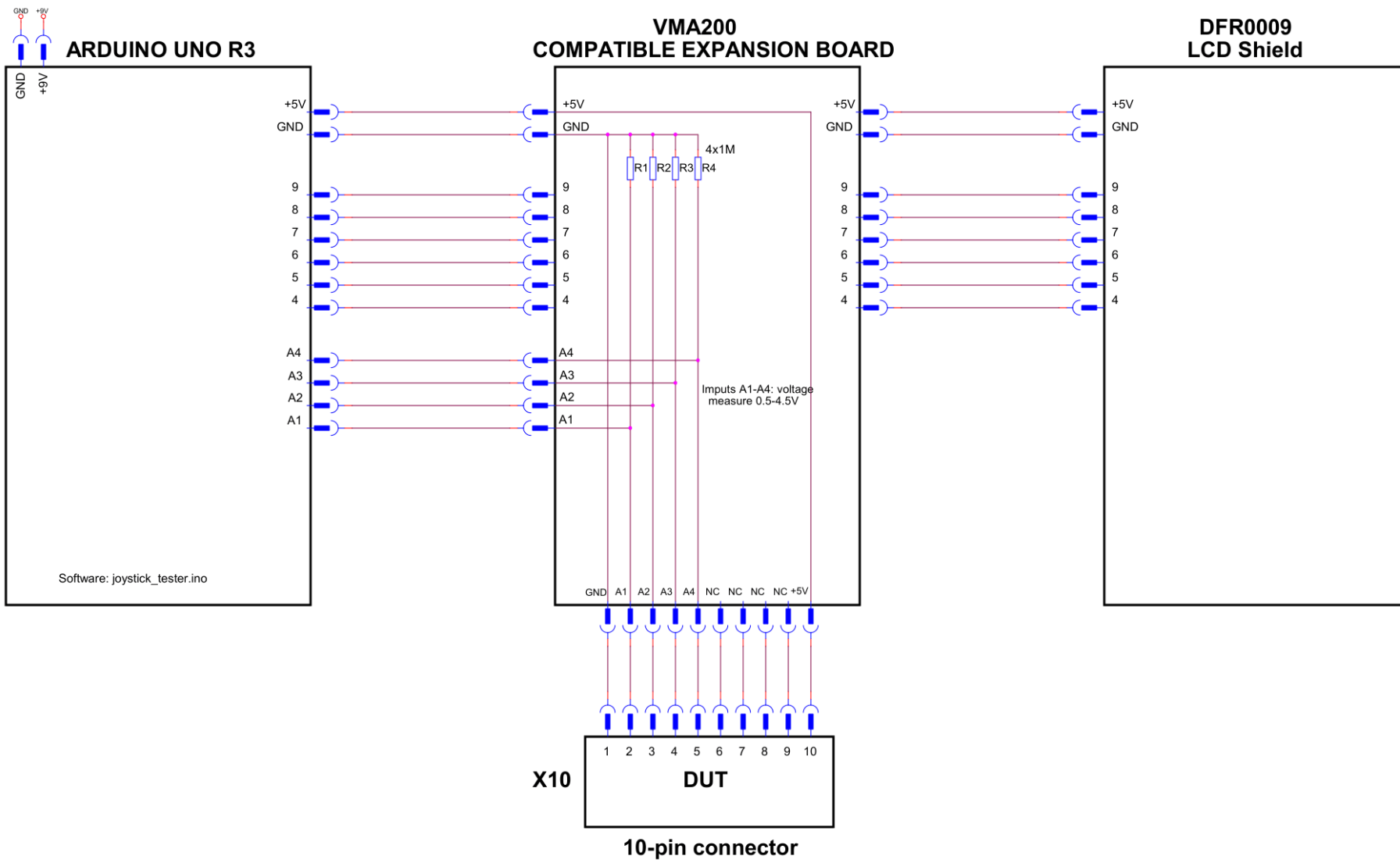
Příloha B: Finální schéma zapojení Testeru univerzálních řídicích desek



Příloha C: Graf průběhu teplot odlitku, vzduchu v komoře a motoru



Příloha D: Schéma zapojení Joystick testeru



Příloha E: Schéma zapojení Foil testeru

